NAWCWD TP 8472

Level Set Methods and Their Application to the Problem of Finding Roads and Rivers in Imagery

by

Phuc Tran

and

Jack Van Wieren

Research Department

March 2000

NAVAL AIR WARFARE CENTER WEAPONS DIVISION
CHINA LAKE, CA 93555-6100

Approved for public release; distribution is unlimited.

Naval Air Warfare Center Weapons Division

FOREWORD

This report documents work carried out and completed during fiscal year 1999 under a Naval Air Warfare Center Weapons Division Core Science and Technology discretionary program, and was funded though the Image and Signal Processing Science and Technology Network. These investigations provided a better understanding of level set image processing and of its potential for finding roads and rivers.

This report was reviewed for technical accuracy by Gary Hewer.

Approved by
JOHN W. FISCHER, *Head*
Research Department
31 March 2000

Under authority of
CHARLES H. JOHNSTON, JR
RDML, U.S. Navy
Commander

Released for publication by
KAREN HIGGINS
Director for Research and Engineering

NAWCWD Technical Publication 8472

Published by . . . . Technical Information Division
Collation . . . . Cover, 16 leaves
First printing . . . . 37 copies

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 2000	Fiscal Year 1999

4. TITLE AND SUBTITLE
LEVEL SET METHODS AND THEIR APPLICATION TO THE PROBLEM OF FINDING ROADS AND RIVERS IN IMAGERY (U)

5. FUNDING NUMBERS

6. AUTHOR(S)
Phuc Tran and Jack Van Wieren

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Air Warfare Center Weapons Division
China Lake, CA 93555-6100

8. PERFORMING ORGANIZATION REPORT NUMBER
NAWCWD TP 8472

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Naval Air Warfare Center Weapons Division
China Lake, CA 93555-6100

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12A. DISTRIBUTION/AVAILABILITY STATEMENT
A Statement; public release; distribution unlimited.

12B. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

(U) This report describes investigations carried out during FY99 with limited discretionary funds. The purpose of this effort was to explore level set image processing methods involving flow under curvature, with the goal of discovering an effective algorithm for finding roads and rivers in gray-scale imagery. Because flow under curvature produces greatest effects with high curvature and least effects with low curvature features such as roads and rivers, taking advantage of the discrimination that this affords became the focus of the work. The Min/Max flow approach was utilized throughout this study.

(U) To develop familiarization, standard problems of notch and noise removal with black-and-white images were studied. It is shown that isolated noise cannot be removed by the unmodified Min/Max flow algorithm, and a successful modification is described and implemented.

(U) The effects of Min/Max flow on various shapes and orientations is then studied experimentally and results are described. Although smaller features disappear rapidly as hoped, there is also erosion in essentially 0 curvature features due to their discrete representation. As a result, some hybrid approaches combining level set methods with some alternative segmentation algorithms are explored. Some promising results were obtained which point to the need for further work.

14. SUBJECT TERMS
Image Processing
Level Set Methods
Segmentation
Algorithms

15. NUMBER OF PAGES
33

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18
298-102

CONTENTS

INTRODUCTION

This technical report details our study of "Using Level Set Methods for Identifying and Following Roads." The Image and Signal Processing Science and Technology Network, Naval Air Warfare Center Weapons Division, China Lake, Calif., funded this project in fiscal year 1999. Our study begins in the next section, "Understanding Level Set Methods" (notch and noise removal), with an effort to obtain a deeper understanding of level set algorithms (see for example Reference 1). In order to do so, we wrote our own level set code to solve the notch removal and noise removal problems. These problems have already been studied by others using the level set method (Reference 2), and so they provide a benchmark to test the correctness of our implementation. We were able to reproduce published results for the notch removal problem but not for the noise removal problem. To help us in understanding these results, we analyzed several simple image cases under level set processing. This is described in the "Analysis of Some Simple Pixel Schemes" section. From these simple analyses we conclude that the level set algorithm cannot remove isolated noise pixels because at an isolated noise pixel, the curvature (as approximated by a central differencing scheme) is undefined. In the "Alternative Scheme To Evaluate the Curvature of an Isolated Noise Pixel" section, a solution to the noise removal problem is given using alternating forward and backward derivative approximations in the curvature calculations for isolated noise pixels. It should be noted here that this approach was found after the rest of the work was completed and has not yet been extended to the gray-scale image processing with which the remainder of the report is concerned.

The details of our efforts to find roads using the level set algorithm are described in the "Road Finding Using Level Set With Edge Detection" section. Although the "Edge Detection Then Level Set Processing" and "Multichannel Segment and Level Set" sections describe unsuccessful attempts, we feel that one can learn as much from failures as from successes and therefore these were included. The "Road Finding Using Level Set With Edge Detection" section describes a more successful approach, with which we were able to isolate significant road segments in a form that might serve for matching to a template. Entire roads were not yet isolated, however, and there was still some obvious noise that was not removed. Further work is needed to improve on these preliminary results. We should mention that there were additional problems encountered with the level set algorithm that we do not yet understand. One thing in particular that we had trouble with is shock, i.e., a point where the slope is undefined (an example is $exp(-|x|)$ at $x = 0$). Sethian (Reference 1) claims that the level set algorithm handles this correctly, but we had problems using his solutions. This shows up as specks of noise in the image after it undergoes level set processing, as shown later in this report. As this problem does not

affect the road finding application and so was beyond the scope of this project, we did not pursue it further.

Finally, all software was written in C and run on a Silicon Graphics Incorporated Challenge XL computer. Complete program listings are provided at http://rattler.chinalake.navy.mil/~jmv/Level_Set/listing.html.

UNDERSTANDING LEVEL SET METHODS

BACKGROUND

To help us understand the level set algorithm, we begin by applying the level algorithm to two problems that have been studied by Malladi and Sethian (Reference 2) to see if we could reproduce their results. These problems are notch removal and noise removal. Before we show those results we briefly discuss the basic level set equation.

The level set method evolves the image [$I(x,y)$] according to the following equation

$$I_t = F|\nabla I| \tag{1}$$

where F is the speed function. This is Equation 8 of Malladi and Sethian (Reference 2). There are several options for the speed function:

$$F = \kappa \tag{2a}$$

$$F = \min(\kappa,0) \tag{2b}$$

$$F = \max(\kappa,0) \tag{2c}$$

$$F = \begin{cases} \min(\kappa,0) & \text{if } I_{ave} < Threshold \\ \max(\kappa,0) & otherwise \end{cases} \tag{2d}$$

where κ is the curvature defined as

$$\kappa = \nabla \cdot \left(\frac{\nabla I}{|\nabla I|} \right) = \frac{I_{xx} I_y^2 - 2 I_x I_y I_{xy} + I_{yy} I_x^2}{\left(I_x^2 + I_y^2 \right)^{3/2}} \tag{2e}$$

The curvature is evaluated by finite difference approximation to the derivative operator. Equation 1 is integrated in time according to procedures given by Sethian (Reference 1). In Equation 2d, the average intensity is defined over a square of size $(2^*\text{stencil} + 1)^2$ pixels, where *stencil* is an integer. The threshold is taken to be 127.5 for black and white images. For a gray-scale image, the threshold is determined following Malladi and Sethian (Reference 2).

NOTCH REMOVAL

Given an image of a black square, with notches on its side, on a white background, the goal is to use the level set algorithm to remove the notches and restore the square. The original image is shown in Figure 1.

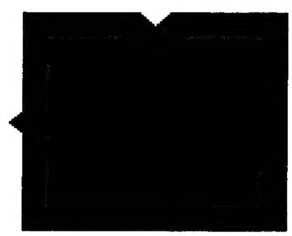


FIGURE 1. Original Image.

Figure 2 shows the image after processing using the speed function given by Equation 2a with dt = 0.1, dx = 1, dy = 1, and 1000 iterations.

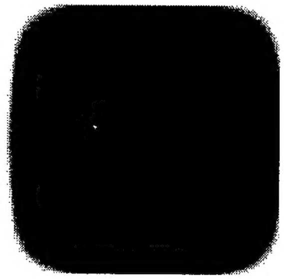


FIGURE 2. Level Set With Speed Function Given by Equation 2a.

Figure 3 shows the image after processing using the speed function given by Equation 2b with dt = 0.1, dx = 1, dy = 1, and 1000 iterations.

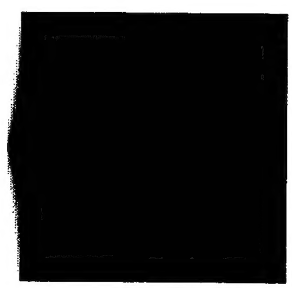


FIGURE 3. Level Set With Speed Function Given by Equation 2b.

Figure 4 shows the image after processing using the speed function given by Equation 2c with dt = 0.1, dx = 1, dy = 1, and 1000 iterations.

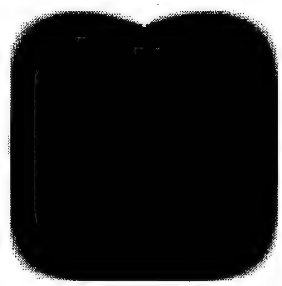


FIGURE 4. Level Set With Speed Function Given by Equation 2c.

Figure 5 shows the image after processing using the speed function given by Equation 2d with dt = 0.1, dx = 1, dy = 1, 1000 iterations, and a stencil size of 3 (remember that the stencil size determines the square size over which the average intensity is calculated).

FIGURE 5. Level Set With Speed Function Given by Equation 2d and Stencil of 3.

The notches are still visible, but that is a function of the stencil size. Figure 6 shows the results with stencil size of 5.

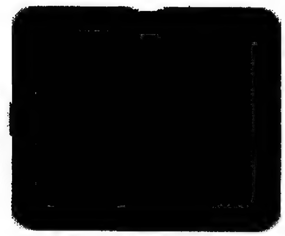
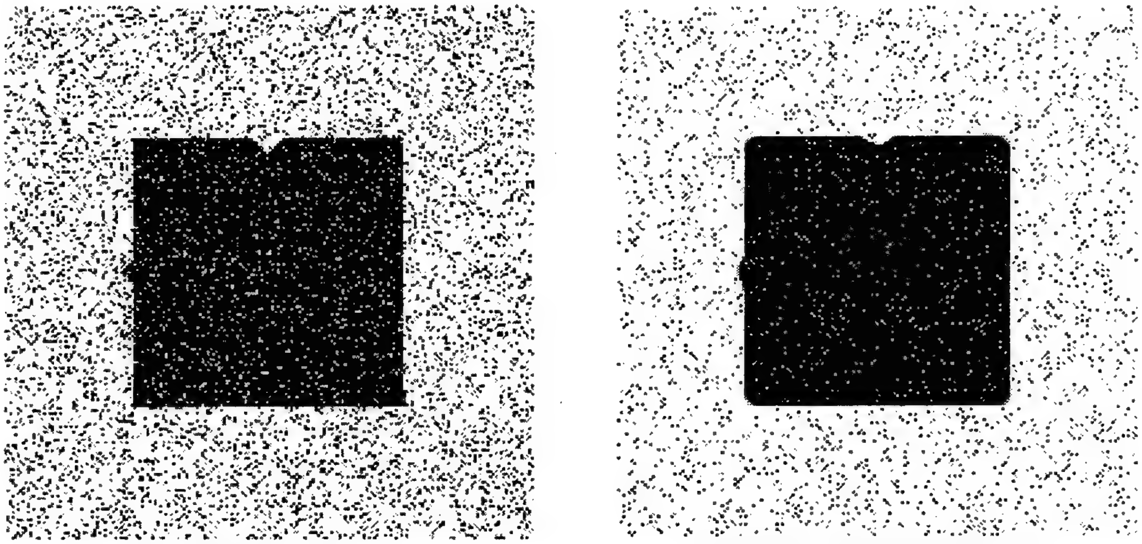


FIGURE 6. Level Set With Speed Function Given by Equation 2d and Stencil of 5.

The above results agree with those of Malladi and Sethian (Reference 2).

NOISE REMOVAL

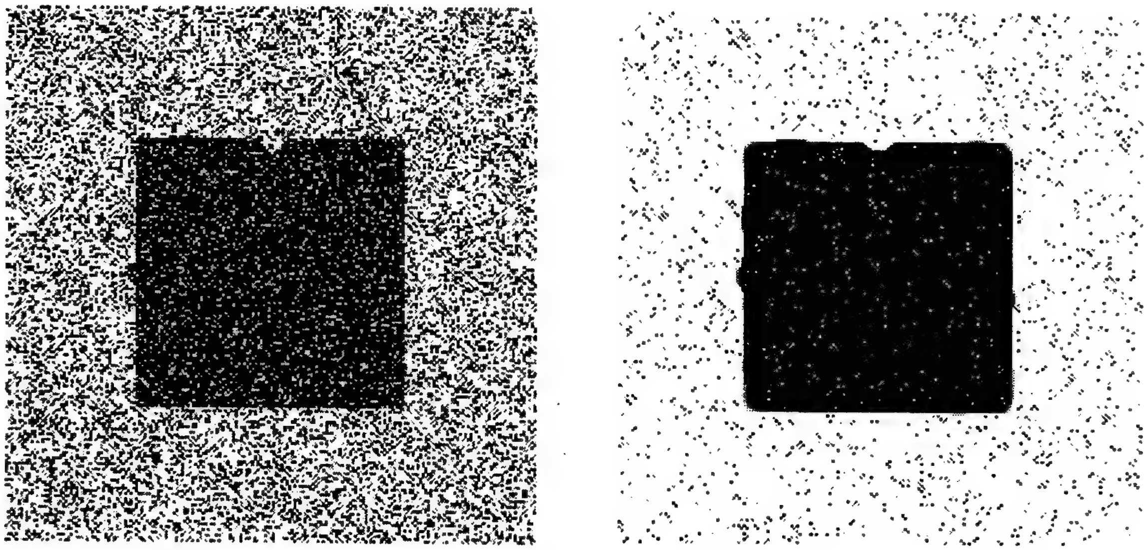
We then studied the noise removal problem. For this problem we use the speed function given by Equation 2d. We take the image in Figure 1 and add various levels of random gray-scale noise. The procedure for adding the random noise is as follows. For a given level of noise (for example, 20% (0.2)), we pick a random number between 0 and 1 for each pixel. If the random number is less than the noise level, then that pixel value is replaced by a random value chosen between 0 and 255. Figure 7a shows the image with 20% noise level, and Figure 7b shows the results after processing with the level set algorithm (1000 iterations).



(a) Original image. (b) Post processing image.

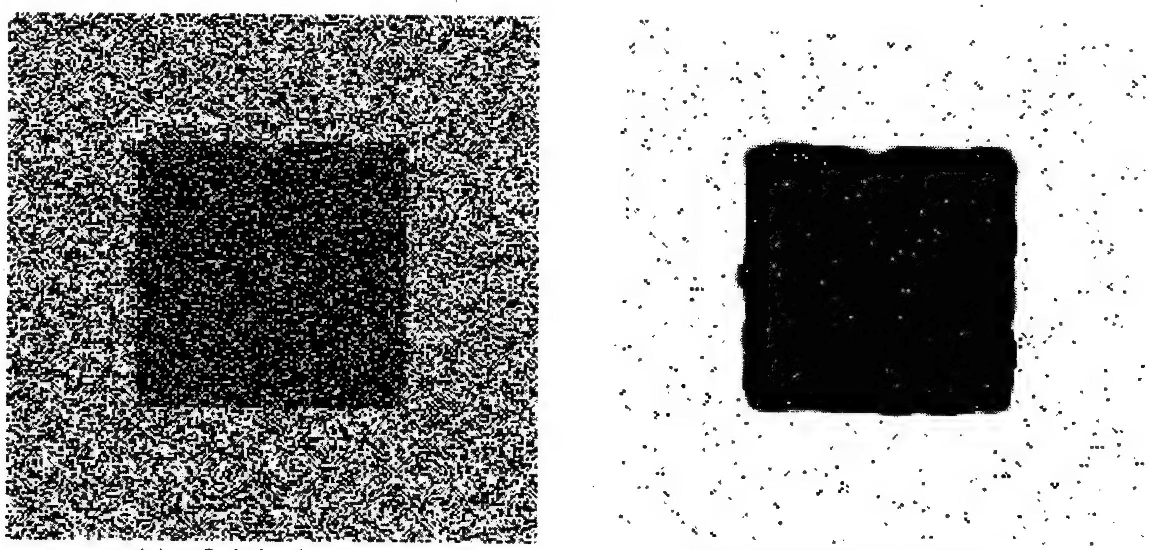
FIGURE 7. Image With 20% Noise.

Figures 8 and 9 show the results for 40 and 60% noise level, respectively. There are two observations to make from these results: (1) the noise is not removed completely, and (2) the more noise there is, the better the noise removal. This is not the same result that Malladi and Sethian reported (Reference 2). Their results show a smooth gray background with no noise speckle. In the "Analysis of Some Simple Pixel Schemes" section, we analyze some simple images that yield understanding of our results so far.



(a) Original image. (b) Post processing image.

FIGURE 8. Image With 40% Noise.



(a) Original image. (b) Post processing image.

FIGURE 9. Image With 60% Noise.

ANALYSIS OF SOME SIMPLE PIXEL SCHEMES

To analyze some simple images, we need the finite difference expression for the curvature. The curvature κ is defined as

$$\kappa = \nabla \cdot \left(\frac{\nabla I}{|\nabla I|} \right) = \frac{I_{xx} I_y^2 - 2 I_x I_y I_{xy} + I_{yy} I_x^2}{\left(I_x^2 + I_y^2 + \varepsilon \right)^{3/2}} \tag{3}$$

where $I_q = \partial I / \partial q$ and ε is a small constant to avoid division by zero. Using the central difference scheme, we have the following formulas for the derivatives needed to calculate the curvature

$$\frac{\partial I(i,j)}{\partial x} = I_x(i,j) = \frac{I(i+1,j) - I(i-1,j)}{2\Delta x} \tag{4}$$

$$\frac{\partial I(i,j)}{\partial y} = I_y(i,j) = \frac{I(i,j+1) - I(i,j-1)}{2\Delta y} \tag{5}$$

$$\frac{\partial^2 I(i,j)}{\partial x^2} = I_{xx}(i,j) = \frac{I(i+1,j) - 2I(i,j) + I(i-1,j)}{\Delta x^2} \tag{6}$$

$$\frac{\partial^2 I(i,j)}{\partial y^2} = I_{yy}(i,j) = \frac{I(i,j+1) - 2I(i,j) + I(i,j-1)}{\Delta y^2} \tag{7}$$

$$\frac{\partial^2 I(i,j)}{\partial x \partial y} = I_{xy}(i,j) = \frac{I(i+1,j+1) - I(i+1,j) - I(i,j+1) + I(i-1,j-1)}{4\Delta x \Delta y} \tag{8}$$

With the above formulas we can examine the behavior of some image features. But before we can proceed, we need to define how to calculate I_{ave}. It is calculated by taking a square with side equal to (2*stencil+1) pixels. The average value of the pixel is calculated for that square.

White Notch in Black Background

Let us look at the evolution of the 4 numbered pixels in Figure 10.

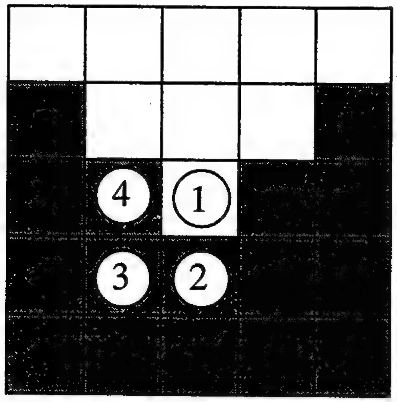


FIGURE 10. White Notch.

Pixel #1:

$I_x = 0$, $I_y = 255/2$, and $I_{xx} = -255$ (note that white is 255, black is 0, and we take $\Delta x = \Delta y = 1$). The curvature κ therefore is < 0. For any stencil > 0, we have $I_{ave} <$ threshold $= 127.5$, therefore $F = \min(\kappa, 0) = \kappa$. Putting everything together for pixel #1 we have

$$I(t+\Delta t) = I(t) + \kappa|\nabla I(t)|\Delta t$$

and because $\kappa < 0$, the value for pixel #1 is getting smaller (i.e., it is becoming black).

Pixel #2:

$$I_x = I_{xx} = 0$$

therefore $\kappa = 0$. Hence

$$I(t+\Delta t) = I(t)$$

No change for pixel #2.

Pixel #3:

$$I_x = I_y = 0$$

therefore $\kappa = 0$. Hence

$$I(t + \Delta t) = I(t)$$

No change for pixel #3.

<u>Pixel #4:</u>

$$I_x = 255/2 \quad , \quad I_{xx} = 255$$

$$I_y = 255/2 \quad , \quad I_{yy} = 255$$

$$I_{xy} = 255/4$$

$$\kappa > 0$$

$$I_{ave} < 127.5$$

So $F = \min(\kappa, 0) = 0$.

$$I(t + \Delta t) = I(t)$$

No change for pixel #4.

SO THE BLACK REGION IS EXPANDING WHILE THE WHITE REGION IS CONTRACTING.

Black Notch in White Background

After the last example, we can just look at pixel #1 in Figure 11 to see which region is expanding and which region is contracting.

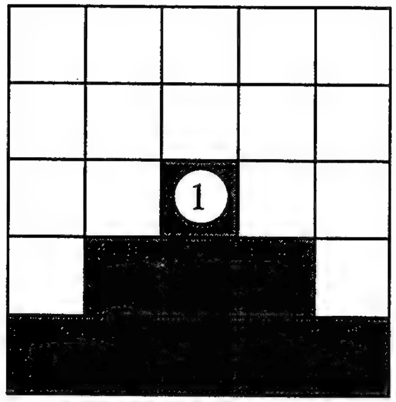


FIGURE 11. Black Notch.

$I_x = 0$, $I_{xx} = 2(255)$

$I_y = 255/2$, $I_{yy} = 255$

$\kappa > 0$

$I_{ave} > 127.5$

therefore $F = \max(\kappa, 0) = \kappa$. Hence

$$I(t+\Delta t) = I(t) + \kappa|\nabla I(t)|\Delta t$$

Pixel #1 is increasing in value or becoming whiter. THE WHITE REGION IS EXPANDING WHILE THE BLACK REGION IS CONTRACTING.

The above analysis agrees with the notch removal simulation given in the "Notch Removal" section of this report and shown again in Figure 12.

(a) Original square with notches.

(b) Image after level set processing.

FIGURE 12. Original Square With Notches and Image After Level Set Processing.

A Black Pixel in a White background

For this case we only need to look at pixel #1 and those pixels surrounding it (see Figure 13). Because of symmetry, we only need to look at pixel #2 and #3.

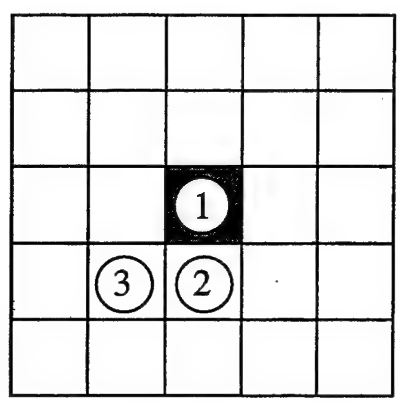


FIGURE 13. Black Pixel (Noise) in White Background.

Pixel #1:

$$I_x = 0 \quad , \qquad I_y = 0$$

Therefore κ is 0, and the pixel does not change (**note**: κ is actually undefined because the denominator is also 0, but we add a small ε to the denominator to keep from dividing by 0. Therefore κ is 0. This points out the deficiency of the central differencing scheme for evaluating κ of a noise pixel and why it does not work).

Pixel #2:

$$I_x = 0 \quad , \qquad I_{xx} = 0$$

Therefore κ is 0, and the pixel does not change.

Pixel #3:

$$I_x = 0 \quad , \qquad I_y = 0$$

Therefore κ is 0, and the pixel does not change.

NOTE THAT THIS IS INDEPENDENT OF THE VALUE OF THE BLACK PIXEL. We get the same results for a white pixel in a black background.

How Does the Above Analysis Help Us to Understand the Simulation Results?

Let us take the square with notches in it as before. Suppose we take one pixel at random and give it a random value. If this pixel is not at the edge of a black and white region, there is no way to remove this "noise" pixel by running the level set algorithm (this follows from the above analysis).

Now suppose we randomly replace two pixels instead of one. Again there is no way to remove the noise, unless the two randomly chosen pixels are near each other. Then we have to do the analysis for such a case. However, the point is that isolated noise pixels cannot be removed.

What this implies is that the more noise we have, the better it can be removed because the probability of clustering is greater. This is in fact what we see in the simulation.

Our problem can be traced back to the fact that curvature is undefined for an isolated noise pixel. In the next section we describe an alternating forward and backward approximation that will eliminate the undefined curvature and will allow complete noise removal.

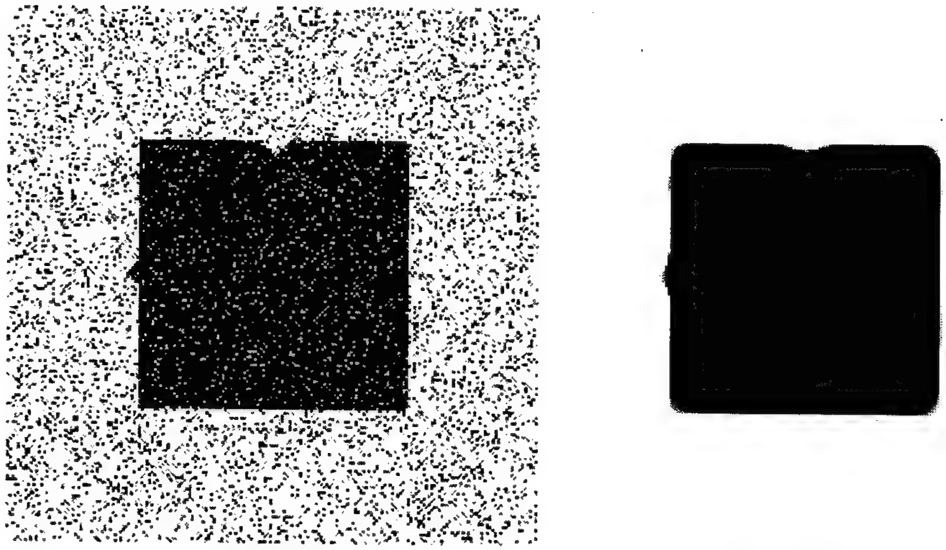
ALTERNATIVE SCHEME TO EVALUATE THE CURVATURE OF AN ISOLATED NOISE PIXEL

Because the curvature, as calculated using the central difference approximation, is undefined for an isolated noise pixel, we use an alternating forward and backward difference approximation. This was suggested by Osher and others.* This procedure is given below (written in C):

```
if ( grad(a(i,j)) = 0)                          " a(i,j) is the 2D image"
        {
          switch (it % 4)                       "it  is the time iteration count"
          {
            case 0:
              da/dx = (a[i+1][j] - a[i][j])/Δx ;     "Forward differencing"
              da/dy = (a[i][j+1] - a[i][j])/Δy ;     "Forward differencing"
              break ;
            case 1:
              da/dx = (a[i][j] - a[i-1][j])/ Δx ;    "Backward differencing"
              da/dy = (a[i][j] - a[i][j-1])/ Δy ;    "Backward differencing"
              break ;
            case 2:
              da/dx = (a[i+1][j] - a[i][j])/ Δx ;    "Forward differencing"
              da/dy = (a[i][j] - a[i][j-1])/ Δy ;    "Backward differencing"
              break ;
            case 3:
              da/dx = (a[i][j] - a[i-1][j])/ Δx ;    "Backward differencing"
              da/dy = (a[i][j+1] - a[i][j])/ Δy ;    "Forward differencing"
              break ;
          }
```

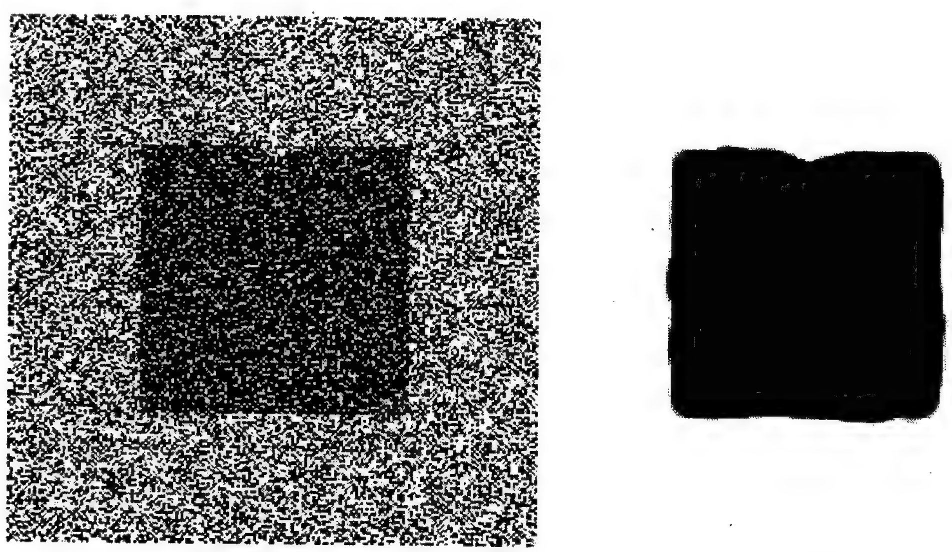
The second derivatives needed to calculate the curvature could be calculated using central differencing. Using this scheme we were able to remove the noise completely. Figures 14 and 15 show the new results for the 20 and 60% noise problem, respectively.

* Personal communication with the authors, 1999



(a) Before. (b) After.

FIGURE 14. 20% Noise dt = 0.1, dx = 1, dy = 1, nt = 100.



(a) Before. (b) After.

FIGURE 15. 60% Noise dt = 0.1, dx = 1, dy = 1, nt = 100.

ROAD FINDING

EDGE DETECTION THEN LEVEL SET PROCESSING

One idea for road finding is to put an image through an edge detector and then apply the level set processing to the output. To see what would happen, we ran the level set code on an image with different kinds of lines (simulating the possible output of the edge detector, see Figures 16 through 19).

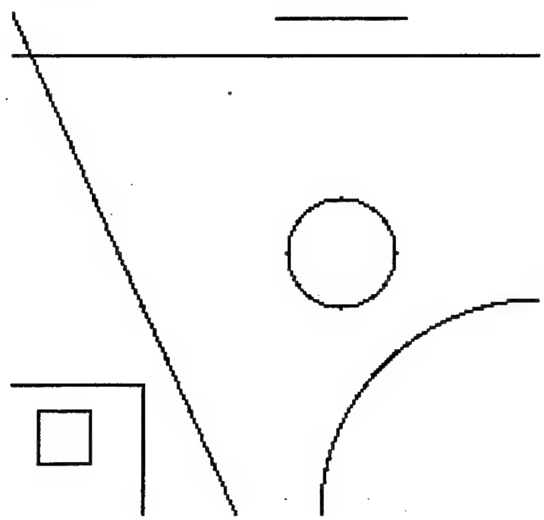


FIGURE 16. Original Image With Various Kinds of Lines.

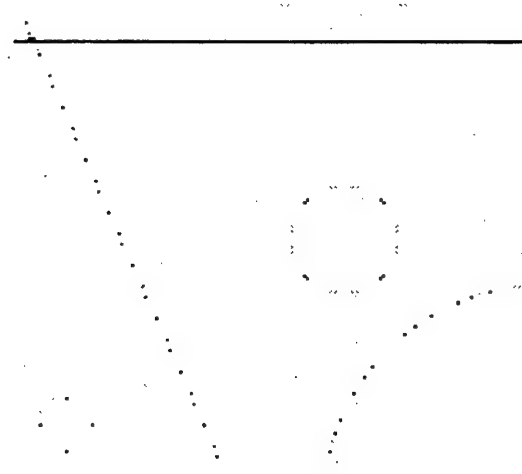


FIGURE 17. Image After 200 Iterations.

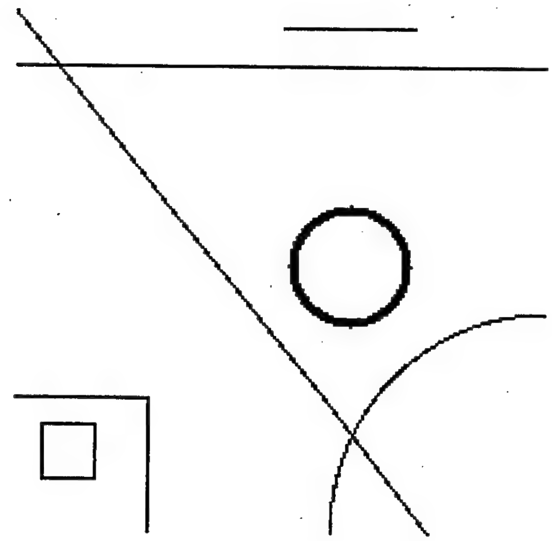


FIGURE 18. Original Image.

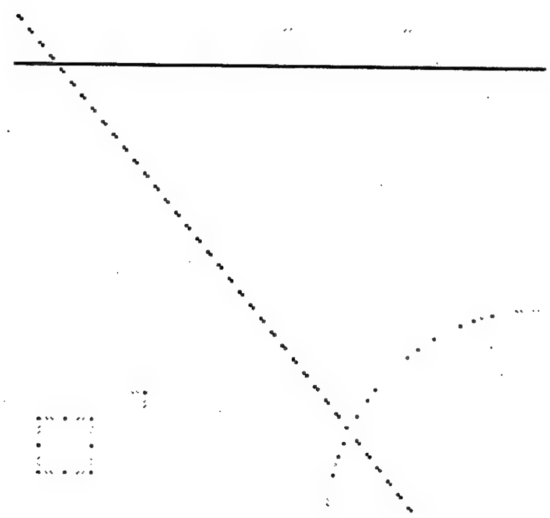


FIGURE 19. Image After 310 Iterations.

From the two previous examples we see that unless the line segment is horizontal or vertical AND traverses from one side to the other, it will not be preserved. THIS IS NOT GOOD FOR ROAD FINDING.

MULTICHANNEL SEGMENTATION AND LEVEL SET

We obtained Jorge Martin's code (Reference 3) to do multichannel segmentation to see if we could use it in conjunction with the level set processing to find roads. The idea was to do segmentation on the image before and after level set processing and then compare the segmented pictures. The road should be the same in both images. The results of this experiment are shown in Figures 20 through 24. Our conclusion is that we get the road, but also a lot of noise that we currently do not know how to eliminate.

FIGURE 20. Original Image.

FIGURE 21. Image After Level Set Processing.

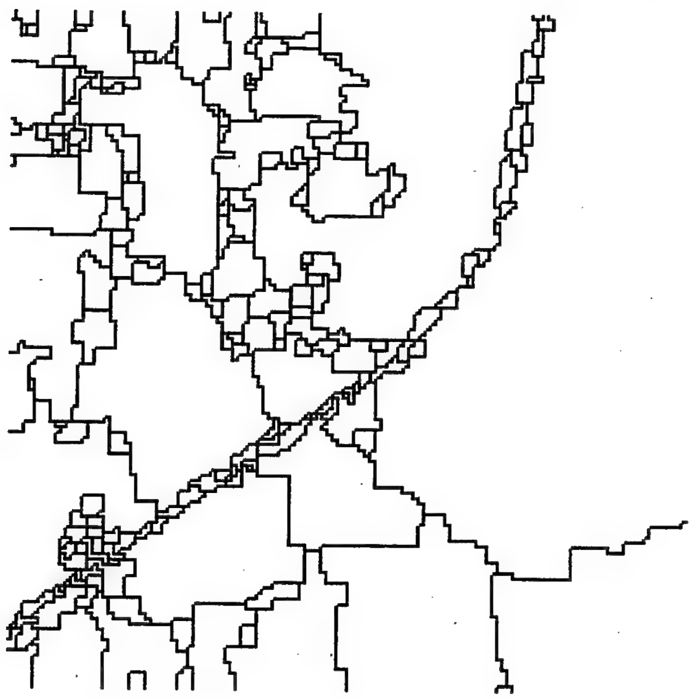


FIGURE 22. Segmentation of Original Image.

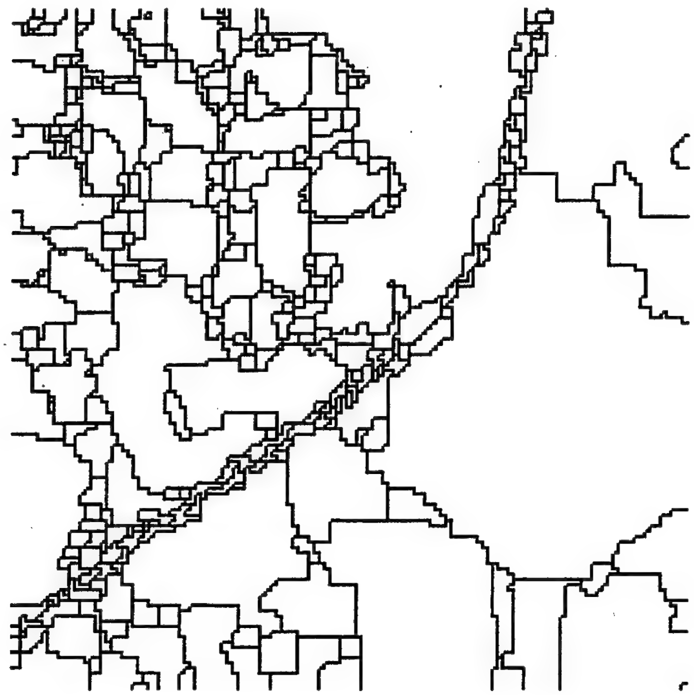


FIGURE 23. Two Channel Segmentation of Original and Post Processing Image.

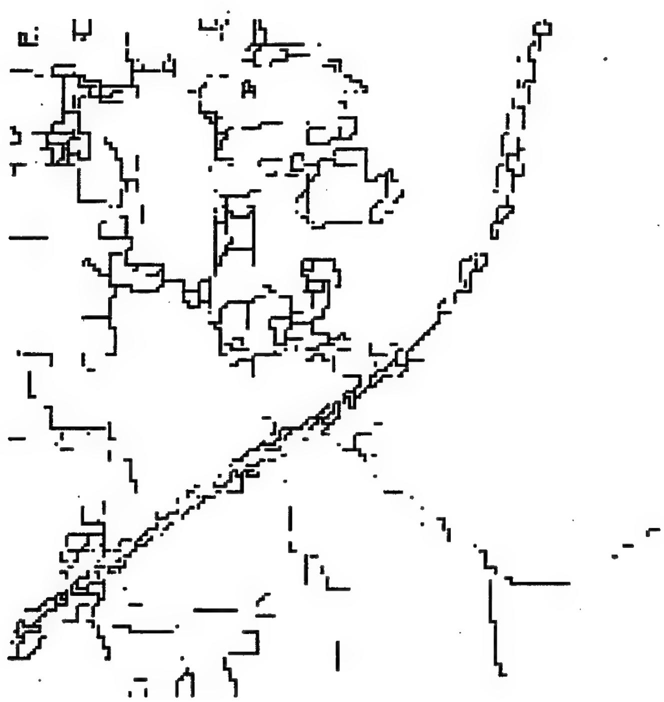


FIGURE 24. "ANDing" of Figures 22 and 23.

ROAD FINDING USING LEVEL SET WITH EDGE DETECTION

This section is a variation on the idea in the "Edge Detection Then Level Set Processing" section. This idea for finding a road by incorporating edge detection is as follows. Because roads have very small curvature, they will not move under the level set algorithm. So if we compare the edges of the original image and the one after level set processing, the common edges will be the roads (see Figures 25 and 26).

FIGURE 25. Original Image.

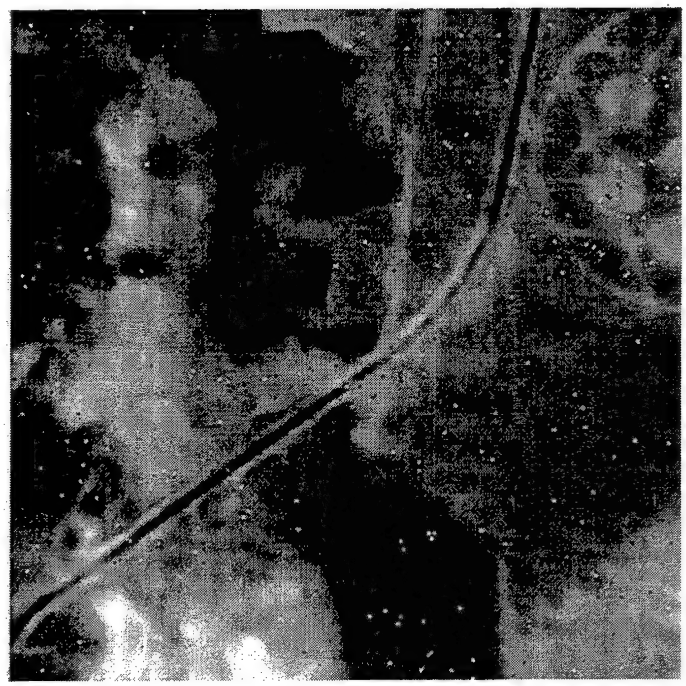


FIGURE 26. Image After Level Set Processing.

For edge detection we used the program of Alan Van Nevel (Reference 4). The results shown are from using the Haar wavelets (see Figures 27 through 29).

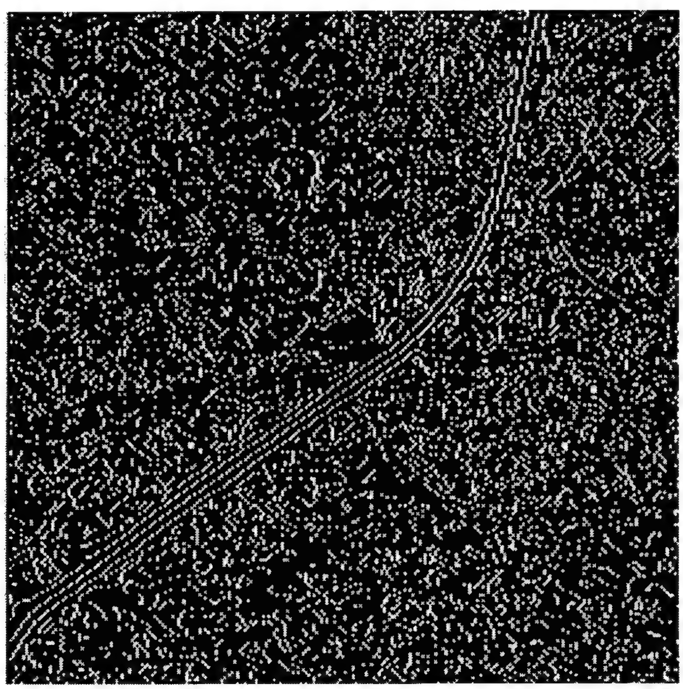


FIGURE 27. Edges From the Original Image.

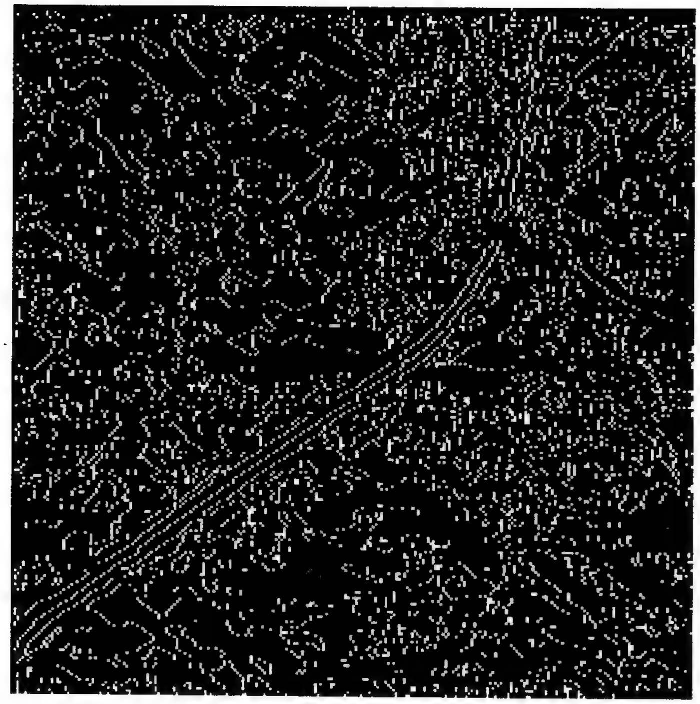


FIGURE 28. Edges From the Post-Processing Image.

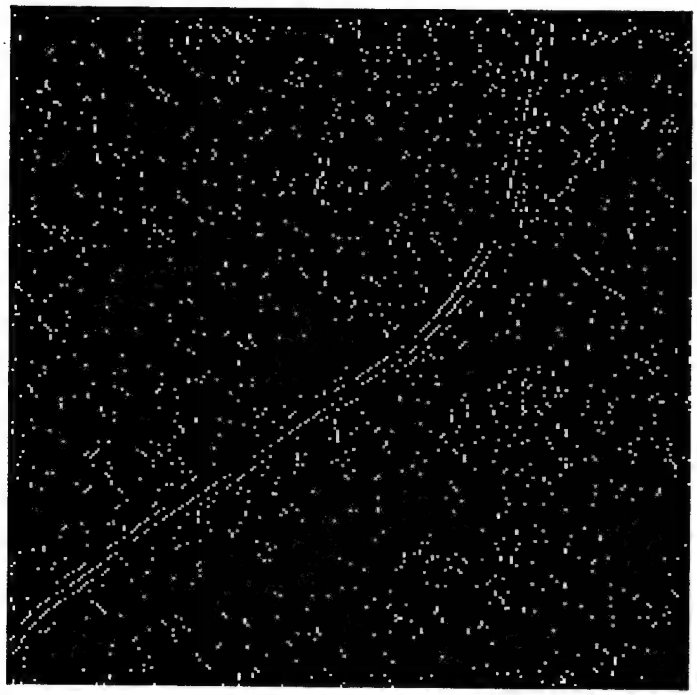


FIGURE 29. "ANDing" of Figures 27 and 28.

Figure 29 shows that our idea is correct, but there is also a lot of noise. However, unlike the situation with the segmentation idea, we can remove the noise. First we eliminate all isolated bright pixels (see Figure 30).

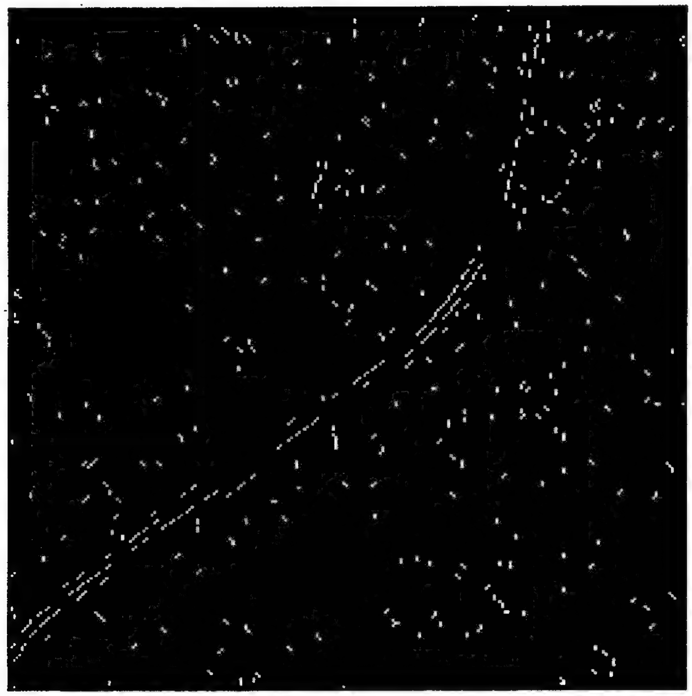


FIGURE 30. Removing Isolated Bright Pixels.

The next step is to remove pairs of bright pixels. However, we note that the road segments are composed of closely spaced pairs of bright pixels. So in order to prevent removal of road segments, we illuminate any dark pixel that is between two bright pixels before removing isolated pairs of bright pixels (see Figures 31 and 32).

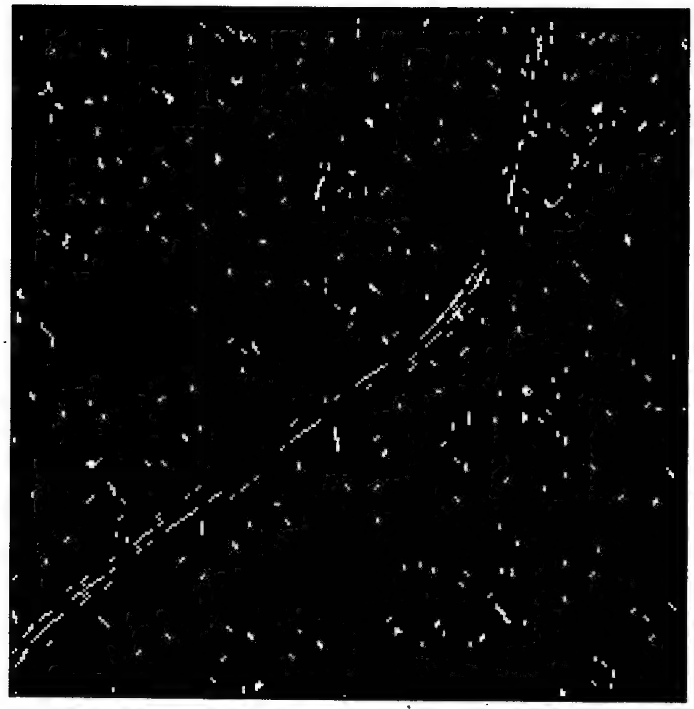


FIGURE 31. Illuminating the Dark Pixel Between Two Bright Pixels.

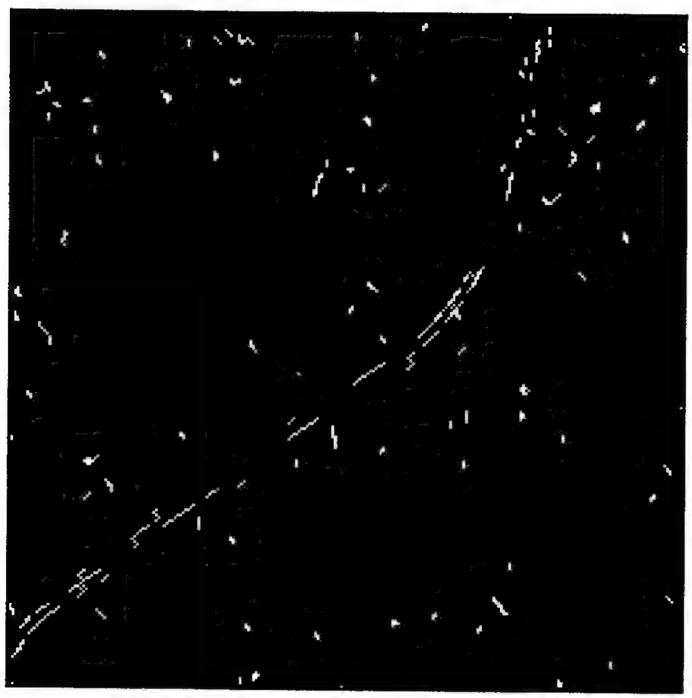


FIGURE 32. Removing Isolated Pairs of Bright Pixels.

We still have some noise after these operations. We can proceed to remove larger groups of pixels but this is not efficient (because there can be many variations of grouping). Because the level set algorithm can collapse to a point feature smaller than a defined stencil size, we will use it to remove the larger grouping. And because the level set algorithm has trouble with small clusters, we enlarge (called "tiling") each bright pixel into a bright square of sizes 3x3 pixels before using the level set algorithm (see Figure 33).

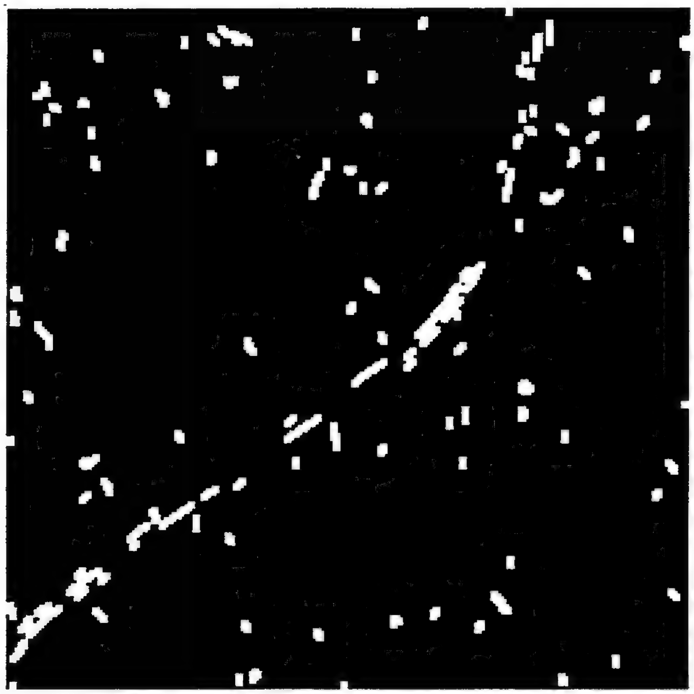


FIGURE 33. Tiling of Figure 32.

The result of tiling (Figure 33) was put through the level set processing for 200 iterations to produce Figure 34. We then performed a thresholding (any pixel less than 127.5 is set to 0) to get Figure 35. Finally, we removed isolated bright pixels again (because the level set algorithm only collapses features into a point). Now we have removed most of the noise, and the image can be used for correlation with a template from a stored map. To see how well this works, we "tiled" the image (Figure 36) using a 7x7 tile to get Figure 37, which was used to mask the original figure (Figure 25). The result is shown in Figure 38.

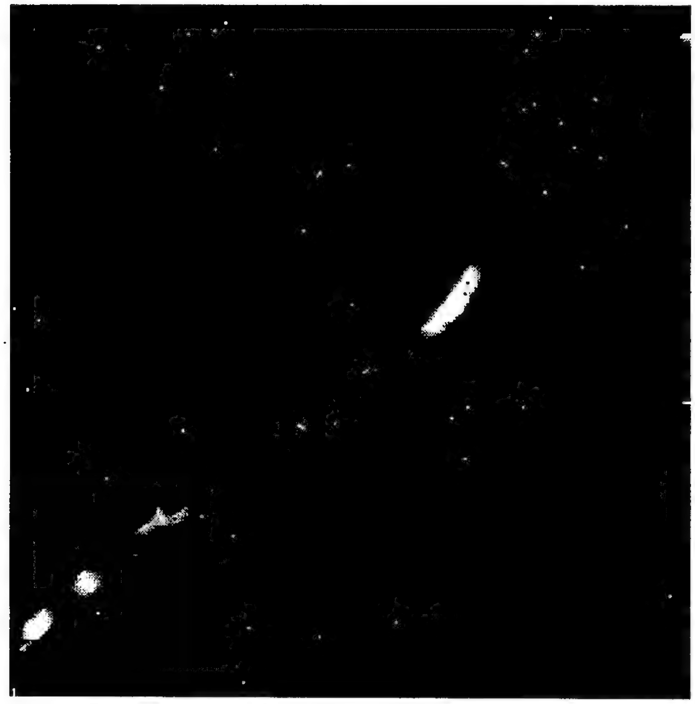


FIGURE 34. After Level Set Processing.

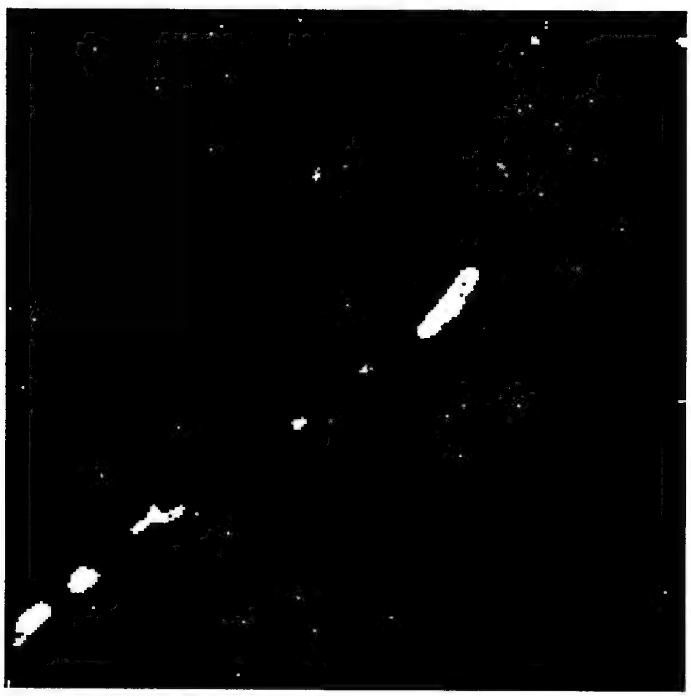


FIGURE 35. Thresholding.

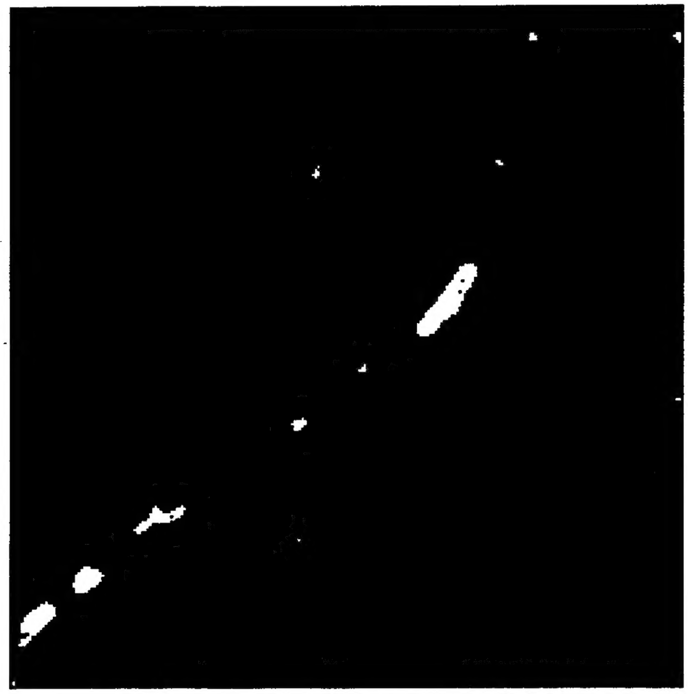


FIGURE 36. Removing Isolated Bright Pixels.

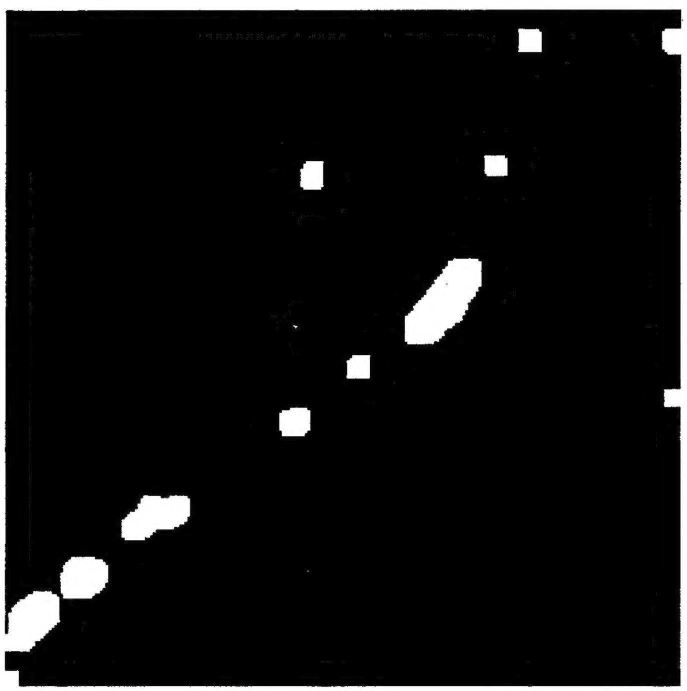


FIGURE 37. Tiling.

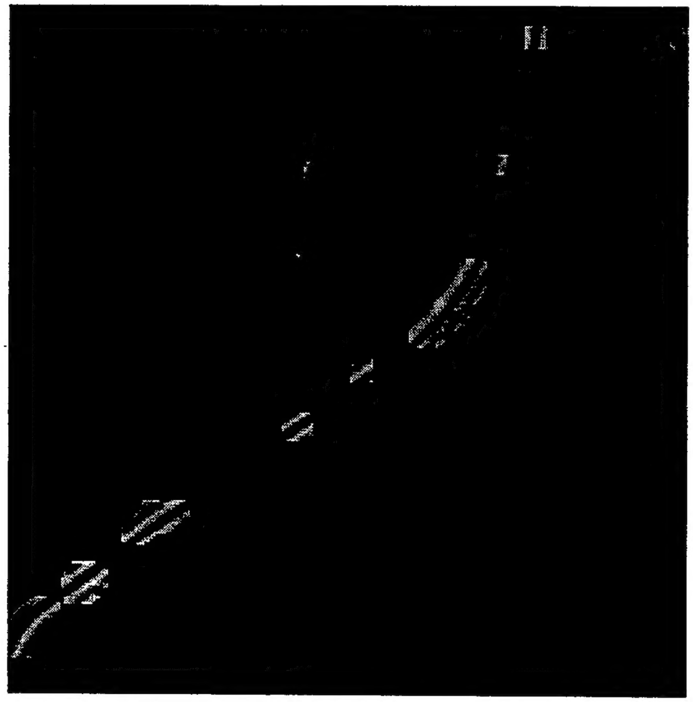


FIGURE 38. Masking.

Figure 38 shows that we can find parts of the road. However some noise still remains. Further work is needed to find more of the road and to eliminate the remaining noise.

REFERENCES

1. J. A. Sethian. *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Materials Science.* Cambridge University Press, 1996.

2. R. Malladi and J. A. Sethian. "Image Processing: Flows Under Min/Max Curvature and Mean Curvature," *Graphical Models and Image Processing*, Vol. 58, No. 2 (1996), pp. 127-141.

3. Naval Air Warfare Center Weapons Division. *A General Merging Criterion*, by J. M. Martin. China Lake, Calif., NAWCWD (in process). (NAWCWPNS TP 8350, publication unclassified.)

4. Alan Van Nevel. "Texture Classification Using Wavelet Frame Decompositions," *31st Asilomar Conference on Signals Systems and Computers, Vol. 2*, ed. by M. P. Fargues and R.D Hippenstiel. Wash., D.C., 2-5 November 1997. (Paper and proceedings unclassified.)

INITIAL DISTRIBUTION

1 Chief of Naval Research, Arlington (Dr. W. Masters)
1 Naval Air Warfare Center Weapons Division, Point Mugu (Code 4T4100E, C. Kenney)
1 Naval War College, Newport (1E22, President)
1 Headquarters, 497 IG/INTW, Falls Church (OUWG Chairman)
2 Defense Technical Information Center, Fort Belvoir
1 Center for Naval Analyses, Alexandria, VA (Technical Library)
1 University of California, Los Angeles, CA (Professor S. Osher)
1 University of California, Santa Barbara, CA (Professor B.S. Manjunath)

ON-SITE DISTRIBUTION

4 Code 4TL000D (3 plus Archives copy)
23 Code 4T4100D
    S. Chesnut (1)
    C. Schwartz (1)
    P. Tran (1)
    J. Van Wieren (20)
1 Code 471600D, G. Hewer